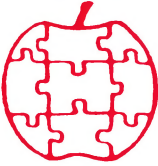


Apple

\$1.80



Assembly

Line

Volume 7 -- Issue 5

February, 1987

In This Issue...

Pre-Allocating DOS Files	2
A Smarter Hexadecimal Memory Search	12
ProVIEW Announcement	19
Display //gs Tool Tables	21
Display the 65802 Registers	25
ProDOS-based Intelligent Disassembler	32

Apple IIgs Technical Reference, by Michael Fischer

Hot off the press, over two pounds and nearly 700 pages of useful reference material for your IIgs. As far as I know, this is the first book to reveal the NUMBERS you need to use the tools in the toolbox. Published by Osborne/McGraw-Hill at \$19.95, our price will be an even \$19. *45 3-1-86*

You know Michael Fischer as the author of 65816/65802 Assembly Language Programming, same publishers. That book is apparently doing well, because McGraw-Hill just raised the price from \$18.95 to \$21.95. Our price went up a little too....

Product Information

Look inside for more detail on two more exciting new products: ProVIEW, by Doug McComsey is a professional tool for zapping your ProDOS disks; the new ProDOS-based S-C Disassembler fills the need for a fast, powerful, and flexible intelligent disassembler operating under ProDOS together with the S-C Macro Assembler.

For years we have been selling the little book by Caxton Foster, "Real Time Programming". It was a little gem, but we cannot get them anymore. The publisher told us they have no schedule for reprinting it. We also miss Foster's excellent book on cryptograms & computers. Both books used the Apple for examples.

Lately I've been working on a project that requires saving large files to disk as fast as possible. The information arrives in 256-byte chunks, with brief pauses between the chunks, so it's not too tough to write to a hard disk during the pauses. I found that things worked fine while saving to an existing file, but when I had to create a new file I got into trouble with the time DOS wastes allocating sectors one by one and constantly reading and re-writing the VTOC. Time for some speedup tricks!

I remembered that the S-C Word Processor uses a special technique to save its files, calling DOS routines to allocate enough sectors and create a track/sector list before actually writing the file. This approach requires that we know how big the file is before opening it, but that's OK because I do have that information. Another problem quickly showed up: the Word Processor knows that its files cannot be bigger than 120 sectors, and therefore will need only one sector's worth of t/u list. My files can be several hundred sectors long, so I have to be ready for anything.

The following machine language routines, along with the Applesoft driver, will create the track/sector lists for a file of any length.

This approach to creating a file does require some non-standard use of DOS routines, so a little patching is necessary. We will be using the File Manager's ALLOCATE.SECTOR routine, which does all that extra VTOC handling. Since we're going to repeatedly call that routine until we're done, it's a waste of time to shuffle the VTOC back and forth. Patching an RTS in at \$AFFD stops the extra VTOC action. Since we're sort of sneaking into the File Manager's territory, we can't rely on the normal error handling. The patch at \$B392 grabs control in the event of an error, so we can remove our patches and handle the error our way.

These programs will almost certainly fail under a modified DOS, so compare the code at the patch locations to standard DOS before trying it out.

Note: Some of the techniques here look odd even to me, but the original program had only a limited space for machine language routines and lots of room for a compiled Applesoft program. That's why, for example, the catalog is updated and rewritten from BASIC in lines 1150-1190, rather than from the machine language.

The Applesoft code is really pretty simple. We issue a DOS UNLOCK command to get the catalog sector containing our file into memory. (If the file doesn't already exist, then the ONERR code at line 10000 creates it.) Once we have the catalog information we can call the machine language routines to 'open' the file by locating the first track/sector list and then allocate the file. Then it's just a matter of looping along collecting data and writing it out to the next sector in the

S-C Macro Assembler Version 2.0.....DOS \$100, ProDOS \$100, both for \$120
 Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....\$20
 ProDOS Upgrade Kit for Version 2.0 DOS owners.....\$30
 Source Code of S-C Macro 2.0 (DOS only).....additional \$100
 Source Code of S-C Macro 2.0 (ProDOS only).....additional \$100
 S-C Disassembler (ProDOS only)...without source code \$30, with source \$50
 ProVIEW (ProDOS only).....\$20
 Full Screen Editor for S-C Macro (with complete source code).....\$49
 S-C Cross Reference Utility.....without source code \$20, with source \$50
 RAK-Ware DISASM (DOS only).....without source code \$30, with source \$50
 S-C Word Processor (with complete source code).....\$50
 DPL8 Source and Object.....\$50
 Double Precision Floating Point for Applesoft (with source code).....\$50
 ES-CAPE (Extended S-C Applesoft Program Editor).....
 Including Version 2.0 With Source Code.....\$50
 ES-CAPE Version 2.0 and Source Code Update (for Registered Owners)....\$30
 Bag of Tricks 2 (Quality Software).....(\$49.95) \$45 *
 MacASM -- Macro Assembler for Macintosh (Mainstay).....(\$150.00) \$50 *
 S-C Documentor (complete commented source code of Applesoft ROMs).....\$50
 Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each
 (Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,
 8051, 8085, 1802/4/5, PDP-11, GI1650/70, others)

AAL Quarterly Disks.....each \$15, or any four for \$45
 Each disk contains the source code from three issues of AAL:
 Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.
 (All source code is formatted for S-C Macro Assembler. Other assemblers
 require some effort to convert file type and edit directives.)

Diskettes (with hub rings)..... package of 25 for \$15 *
 Vinyl disk pages, 6"x8.5", hold two disks each.....10 for \$6 *
 Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each
 (Cardboard folders designed to fit 6"x9" Envelopes.) or \$25 per 100 *
 Envelopes for Diskette Mailers..... 6 cents each

65802 Microprocessor (Western Design Center).....(\$95) \$50 *
 quickLoader EPROM System (SCRG).....(\$179) \$170 *
 PROMGRAMMER (SCRG).....(\$149.50) \$140 *
 Minuteman 250 Uninterruptible Power Supply.....(\$359) \$320 +

"Apple //gs Technical Reference", Fischer.....(\$19.95) \$19 *
 "65816/65802 Assembly Language Programming", Fischer.....(\$21.95) \$20 *
 "Programming the 65816", Eyes.....(\$22.95) \$21 *
 "Apple //e Reference Manual", Apple Computer.....(\$24.95) \$23 *
 "Apple //c Reference Manual", Apple Computer.....(\$24.95) \$23 *
 "ProDOS Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *
 "Apple ProDOS: Advanced Features for Programmers", Little..(\$17.95) \$17 *
 "ProDOS Inside and Out", Doms & Weishaar.....(\$16.95) \$16 *
 "DOSTalk Scrapbook", Weishaar & Kersey.....(\$14.95) \$14 *
 "Beneath Apple ProDOS", Worth & Lechner.....(\$19.95) \$18 *
 "Beneath Apple DOS", Worth & Lechner.....(\$19.95) \$18 *
 "Inside the Apple //c", Little.....(\$19.95) \$18 *
 "Inside the Apple //e", Little.....(\$19.95) \$18 *
 "Understanding the Apple //e", Sather.....(\$24.95) \$23 *
 "Understanding the Apple II", Sather.....(\$22.95) \$21 *
 "Apple II+/IIE Troubleshooting & Repair Guide", Brenner.....(\$19.95) \$18 *
 "Apple II Circuit Description", Gayler.....(\$22.95) \$21 *
 "Assem. Language for Applesoft Programmers", Finley & Myers.(\$18.95) \$18 *
 "Now That You Know Apple Assembly Language...", Gilder.....(\$19.95) \$18 *
 "Enhancing Your Apple II, vol. 1", Lancaster.....(\$15.95) \$15 *
 "Enhancing Your Apple II, vol. 2", Lancaster.....(\$17.95) \$17 *
 "Assembly Cookbook for the Apple II/IIE", Lancaster.....(\$21.95) \$20 *
 "Microcomputer Graphics", Myers.....(\$14.95) \$14 *
 "Assembly Lines -- the Book", Wagner.....(\$19.95) \$18 *
 "AppleVisions", Bishop & Grossberger.....(\$39.95) \$36 *

* On these items add \$2.00 for the first item and
 \$.75 for each additional item for US shipping.

+ Inquire for shipping cost.

Foreign customers inquire for postage needed.

Texas residents please add 6 1/8 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***
 *** (214) 324-2050 ***
 *** Master Card, VISA, Discover and American Express ***

list. 'Closing' the file just means updating the catalog sector with the final length and the correct file type. The routine beginning at line 20000 is just a dummy to come up with something to put in the file.

Lines 1420-1460 provide fixed addresses for BASIC to call the various routines. At line 1520, OPEN.USER.FILE uses the DOS CATALOG.INDEX and the catalog sector in memory to look up the location of the first track/sector list, falling into READ.TS.LIST to get that into memory.

ALLOCATE.FILE, beginning at line 1790, is the heart of this program. We start out by remembering the location of the current t/s list and reading the VTOC. We then install our quick patches into DOS. After initializing the offset, program length (1 because of the first t/s list sector) and index, we're ready to start allocating.

We call NEXT.TS.ENTRY to see if there is already another sector allocated. If so it's on to lines 2060-2140 to count that sector and see if we're done. If NEXT.TS.ENTRY returns with .EQ. status then we need another sector and must call ALLOCATE.SECTOR to get it. That routine returns the sector number in the A-register, so we store that and get the track number from the FM's variable at \$B5F1. We need to decrement the Y-register because NEXT.TS.ENTRY has already moved TS.INDEX to the next position. Once we have the new sector we count it and, if we're not done, go back for more.

Once we have all the sectors we need, we just call the DOS routine to free up the reserved but unused sectors in the current track, clean up our patches, and finally write out the revised VTOC. Then we write out the current t/s list and we're finished.

READ and WRITE.NEXT.SECTOR just set up the IOB, call NEXT.TS.ENTRY to get a sector location, and call RWTS. If an error occurs it is passed back to BASIC for handling. If NEXT.TS.ENTRY return .EQ. status here, then we have hit the end of the file. This feature isn't used in this demo, but is necessary when we use the same techniques to read a file.

NEXT.TS.ENTRY, at line 2550, starts out by getting the current index into the list. Normally that index is greater than zero, so the routine bumps the index for the next call and returns the next sector and track in Y and A. If the index was zero, then we're finished with that sector of the list and go on to line 2640 to see if there is another sector. If there is another sector then we pick up its location, remember whether we're in the middle of reading or writing, and go get the next piece of the list. Then it's back to the top to get the next entry. Note that READ.TS.LIST resets TS.INDEX to the beginning.

If there isn't another sector we drop down to line 2760. If we're not in the middle of allocating then this means we're at the end of the file, so we return with a zero in the A-register. When we are allocating it means we need to make

another t/s list sector. To do this we first call ALLOCATE.SECTOR to find a place for it, then set the pointer bytes in the current sector, recover the current sector's location, and write it out. Then we remember where this sector is going to go, zero the buffer, update the offset, and count the new sector.

FILE.MANAGER.ERROR just cleans out the DOS patches and passes the error code back to the Applesoft program. REMOVE and INSTALL.DOS.PATCHES very straightforwardly do just that.

Much of this code is adapted from the S-C Word Processor's method of reading and writing files. My thanks to Bob S-C and Bobby Deen for the inspiration.

```

10 HIMEM: 37888
20 D$ = CHR$(4)
30 PRINT D$*BLOAD B.PREALLOCATE,A$9400"
40 PA = 37888
100 TYPE = 0: REM Change for Non-Text File
110 EC = 3: POKE EC,0: REM Error Code
120 LN = 3: REM File Length, in Sectors
130 POKE 0,LN - LN / 256: POKE 1,LN / 256
140 F$ = "FILENAME"
1000 ONERR GOTO 10000
1010 PRINT D$*UNLOCK*F$
1020 ONERR GOTO 11010
1030 CALL PA: REM Locate File
1040 IF PEEK (EC) THEN 11000
1050 CALL PA + 3: REM Allocate File
1060 IF PEEK (EC) THEN 11000
1070 PRINT D$*UNLOCK*F$
1080 CALL PA: REM Locate File
1090 IF PEEK (EC) THEN 11000
1100 FOR I = 1 TO LN
1110 GOSUB 20000: REM Get a Sector
1120 CALL PA + 9: REM Write a Sector
1130 IF PEEK (EC) THEN 11000
1140 NEXT
1150 PRINT D$*UNLOCK*F$
1160 CL = PEEK (45980) + 46311
1170 POKE CL, PEEK (4): POKE CL + 1, PEEK (5): REM Catalog File Length
1180 POKE PEEK (45980) + 46280,TYPE: REM If New, Non-Text File
1190 CALL PA + 12: REM Rewrite Catalog Sector
1200 IF PEEK (EC) THEN 11000
1210 END
10000 POKE 216,0
10010 ER = PEEK (222): IF ER < > 6 THEN 11020
10020 PRINT D$*OPEN*F$: PRINT D$*CLOSE"
10030 GOTO 1020
11000 ER = PEEK (EC): GOTO 11020
11010 ER = PEEK (222)
11020 PRINT "ERROR #"ER
11030 POKE 216,0
12000 END
20000 FOR Z = 0 TO 255
20010 POKE 39590,A + 193
20020 NEXT
20030 A = A + 1
20040 RETURN

```

7 Meg Iie/1 Meg Iic

Want the best & easiest-to-use (most support) RAM card on the market — at the best price? Buy Checkmate Technology's State-of-the-Art **MULTIRAM RGB RAM CARD™** from us w/**BATTERY BACKED-UP RAM** options that save programs like **AppleWorks** for 10+ years! **FAST CHEAP REPLACEMENT FOR HARD DISKS, USER EXPANDABLE TO 7 MEGABYTES**, compatible with all (100%) 3rd party software/hardware, optional 65C816 Co-Processor card (\$147), sharp 80 columns/Double Hi-Res, & **FREE RGB!** It's a direct substitute for Ramworks I-III™ by Applied Engineering (also known as Preferred Computing) or Apple 80 col cards, has a **5 YR WARRANTY, 15 day money back guarantee, & FREE AUTOMATIC SOFTWARE UPDATES ONLY FROM US!**

MultiRam RGB expands to 1 Meg RAM + optional 4 Meg's **MultiRam Plus BATTERY PROTECTED RAM**. Optional 2 Meg's **STATIC RAM** here soon — for a total of 7 megabytes! **MultiRam Iie** expands to 768k. And the software & manual is the best on the market!

FREE APPLEWORKS EXPANDER SOFTWARE WITH EVERY CARD — loads ALL versions of Appleworks (in 30 seconds w/AW 2.0), even printer routines, runs faster, expands Desktop over 2048k, stores 23,000+ records & 4,800+ word processor lines, auto-segments large files onto multiple disks! **FREE AW TIME/DAY/DATE ON-SCREEN w/ clock, AUTO-COPY TO RAM, PRODOS/DOS 3.3. RAMDISK & RAM TEST.** Optional CP/M & Pascal Ram disks (\$20/ea)!

	MultiRam RGB Card	MultiRam Iie Card
64k MultiRam	169.	129.
320k MultiRam	199.	164.
576k MultiRam	229.	199.
832k MultiRam	254.	234.
1024k MultiRam	279. ← ← ← ←	

	MultiRam Plus Piggyback	
256k MultiRam Plus ...	220.	NEW
512k MultiRam Plus ...	269.	NEW
768k MultiRam Plus ...	316.	NEW
1024k MultiRam Plus ...	359.	NEW
2-4 meg MultiRam Plus ...	Call	NEW
Battery/AC Kit (for Plus)	36.	NEW

Apple Iie Enhancement Kit	62.
Cermetek 1200 Baud internal modem Iie/II+ /IlgS	179.
Incomm 2400 Baud external univ. modem	319.
Clockworks Clock (Thunder/Time HO™ compat)	89.
13" RGB Monitor (free cable)	322.
ProAPP 10/20 Meg Hard Disks (+ FREE software)	Call
Apple Compatible Drives Iie/II+ (IlgS/Iic +15)	135.
NETWORKING UP TO 256 APPLE Iic/Iie/IlgS	Call

Terms: Add \$4-Ground or \$6-Air shipping & phone # to each U.S. MultiRam card order (foreign orders/FPO/APO extra). Add 3% for MasterCard/Visa (include #/expir) & P.O.'s. For fast delivery send Cashier's/Certified check/Money Order. C.O.D. (add \$6) & personal checks accepted (allow 16 days). Tex res add 7 1/4% tax.

Checkmate Technology's MultiRam CX card easily expands your Iic to 640k and has a **CX+ Piggyback** to add another 512k (over 1152k total)! It's 100% compatible with all Iic software/hardware & upgradable with a 65C816 kit (\$105) — all using about 50% less power than Z-RAM! It has the **SAME FREE SOFTWARE/UPDATES & 5 YR WARRANTY AS Iie CARDS.**

256k MULTIRAM CX	179.
512k MULTIRAM CX	209.
768k MULTIRAM CX Combo	328.
1024k MULTIRAM CX Combo	354.
256k MULTIRAM CX+	210.
512k MULTIRAM CX+	222.

AutoWorks (Best AppleWorks Macro+)	37.
1200 Baud Iic Modem (+\$20/cable)	159.
Clip-on Iic Rechargeable Battery	124.
Iic Flat Panel Display (C-Vue)	299.
Iic System Clock	66.

WHY BUY FROM COIT VALLEY COMPUTERS RATHER THAN SOME MAIL ORDER HOUSES? Only we offer a **15 day MultiRam complete money back satisfaction guarantee**, extra software, more support, **free automatic software updates**, free 64k w/ each 256k/512k/768k Iie/RGB card. We know the products, & stock them for fast shipping! **CALL FOR QUESTIONS, PRICES, DISCOUNTS. SCHOOLS & GROUPS WELCOME.**

Autoworks, Ramworks III/Timeaster II HO /Ilg, register trademarks of Software Tools by Applied Engineering. *Call for RGB Connectors/cables (not included).

ORDER NOW!

COIT VALLEY COMPUTERS (214) 234-5047
14055 Waterfall Way Dallas, Texas 75240

NAME		
ADDRESS		
CITY	STATE	ZIP
PHONE ()		
SIGNATURE		
QTY	DESCRIPTION	PRICE
MC/VS	EXP	SHIP. TOTAL

COIT VALLEY COMPUTERS
(214) 234-5047

MCI 2969684
Telex 6502969684 (WUJ)

14055 Waterfall Way
Dallas, Texas 75240

```

1000 *SAVE S.PREALLOCATE
1010 *-----
00- 1020 COUNT .EQ 0,1
02- 1030 EOF.FLAG .EQ 2
03- 1040 ERROR.FLAG .EQ 3
04- 1050 PGM.LENGTH .EQ 4,5
06- 1060 TS.INDEX .EQ 6
07- 1070 TS.OFFSET .EQ 7,8
09- 1080 THIS.TS.TRACK .EQ 9
0A- 1090 THIS.TS.SECTOR .EQ $A
0B- 1100 ALLOCATE.FLAG .EQ $B
1110
03D9- 1120 CALL.RWTS .EQ $3D9
03E3- 1130 GET.IOB .EQ $3E3
1140
9AA6- 1150 DATA.BUFFER .EQ $9AA6
9BA6- 1160 TS.BUFFER .EQ $9BA6
1170
AFF7- 1180 DOS.READ.VTOC .EQ $AFF7
AFFB- 1190 DOS.WRITE.VTOC .EQ $AFFB
B244- 1200 ALLOCATE.SECTOR .EQ $B244
B2C3- 1210 RELEASE.SECTORS .EQ $B2C3
B39C- 1220 CATALOG.INDEX .EQ $B39C
B4C6- 1230 CAT.TRACK .EQ $B4C6 (+ Index)
B4C7- 1240 CAT.SECTOR .EQ $B4C7 "
B5C5- 1250 FM.ERROR .EQ $B5C5
B5F1- 1260 CURRENT.TRACK .EQ $B5F1
1270
B7E8- 1280 IOB .EQ $B7E8
B7EB- 1290 IOB.VOLUME .EQ IOB+3
B7EC- 1300 IOB.TRACK .EQ IOB+4
B7ED- 1310 IOB.SECTOR .EQ IOB+5
B7F0- 1320 IOB.BUFFER .EQ IOB+8,9
B7F4- 1330 IOB.OPCODE .EQ IOB+12
1340 *-----
1350 .OP 65C02
1360 .OR $9400
1370 .TF B.PREALLOCATE
1380 *-----
1390 *
1400 *
1410 * CALLS FROM BASIC
1420 *
9400- 4C 0F 94 1420 CALL.PA JMP OPEN.USER.FILE
9403- 4C 3F 94 1430 CALL.PA.3 JMP ALLOCATE.FILE
9406- 4C A1 94 1440 CALL.PA.6 JMP READ.NEXT.SECTOR
9409- 4C A4 94 1450 CALL.PA.9 JMP WRITE.NEXT.SECTOR
940C- 4C B8 94 1460 CALL.PA.12 JMP RWTS.CALLER
1470 *-----
1480 *
1490 * GET A FILE READY
1500 *
1510 OPEN.USER.FILE
940F- AE 9C B3 1520 LDX CATALOG.INDEX
9412- BD C6 B4 1530 LDA CAT.TRACK,X t/s list track
9415- 8D EC B7 1540 STA IOB.TRACK
9418- BD C7 B4 1550 LDA CAT.SECTOR,X & sector
941B- 8D ED B7 1560 STA IOB.SECTOR
1570
1580 READ.TS.LIST
941E- A9 01 1590 LDA #1 read opcode
9420- 2C 1600 .HS 2C skip 2 bytes
1610 WRITE.TS.LIST
9421- A9 02 1620 LDA #2 write opcode
9423- 8D F4 B7 1630 STA IOB.OPCODE
9426- A9 A6 B7 1640 LDA #TS.BUFFER set buffer
9428- 8D F0 B7 1650 STA IOB.BUFFER
942B- A9 9B 1660 LDA /TS.BUFFER
942D- 8D F1 B7 1670 STA IOB.BUFFER+1
9430- 20 B8 94 1680 JSR RWTS.CALLER
9433- A9 0C 1690 LDA #$C point to first t/s pair
9435- 85 06 1700 STA TS.INDEX
9437- A9 9A 1710 LDA /DATA.BUFFER restore buffer
9439- 8D F1 B7 1720 STA IOB.BUFFER+1
943C- 64 02 1730 STZ EOF.FLAG
943E- 60 1740 RTS

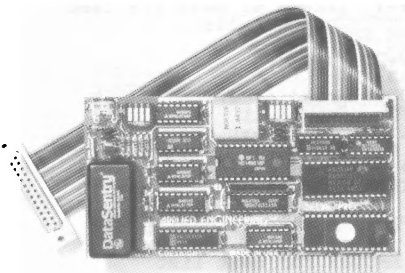
```

```

1750 *-----
1760 *
1770 *       PRE-ALLOCATE SECTORS
1780 *
1790 *       ALLOCATE.FILE
943F- AE 9C B3 1800 LDX CATALOG.INDEX
9442- BD C6 B4 1810 LDA CAT.TRACK,X      remember where
9445- 85 09 1820 STA THIS.TS.TRACK    we're starting
9447- BD C7 B4 1830 LDA CAT.SECTOR,X
944A- 85 0A 1840 STA THIS.TS.SECTOR
944C- 20 F7 AF 1850 JSR DOS.READ.VTOC
944F- 20 46 95 1860 JSR INSTALL.DOS.PATCHES
9452- 64 07 1870 STZ TS.OFFSET
9454- 64 08 1880 STZ TS.OFFSET+1
9456- A9 01 1890 LDA #1
9458- 85 04 1900 STA PGM.LENGTH      count first t/s list
945A- 64 05 1910 STZ PGM.LENGTH+1
945C- A9 0C 1920 LDA #0C             start of t/s pairs
945E- 85 06 1930 STA TS.INDEX
9460- 85 0B 1940 STA ALLOCATE.FLAG  make non-zero
1950
9462- 20 C6 94 1960 .1 JSR NEXT.TS.ENTRY (bumps TS.INDEX)
9465- D0 10 1970 BNE .2             .NE. if one already present
9467- 20 44 B2 1980 JSR ALLOCATE.SECTOR
946A- A4 06 1990 LDY TS.INDEX
946C- 88 2000 DEY
946D- 99 A6 9B 2010 STA TS.BUFFER,Y      set sector
9470- AD F1 B5 2020 LDA CURRENT.TRACK
9473- 88 2030 DEY
9474- 99 A6 9B 2040 STA TS.BUFFER,Y      and track
2050
9477- E6 04 2060 .2 INC PGM.LENGTH      count it
9479- D0 02 2070 BNE .3
947B- E6 05 2080 INC PGM.LENGTH+1
947D- C6 00 2090 .3 DEC COUNT
947F- D0 E1 2100 BNE .1             back for more
9481- A5 01 2110 LDA COUNT+1      are we done?
9483- F0 04 2120 BEQ .4
9485- C6 01 2130 DEC COUNT+1      no, keep going
9487- 80 D9 2140 BRA .1
2150
9489- 20 C3 B2 2160 .4 JSR RELEASE.SECTORS  rest of track
948C- 20 43 95 2170 JSR REMOVE.DOS.PATCHES clean house
948F- 20 FB AF 2180 JSR DOS.WRITE.VTOC  update VTOC
9492- 64 0B 2190 STZ ALLOCATE.FLAG  done allocating
9494- A5 09 2200 LDA THIS.TS.TRACK
9496- 8D EC B7 2210 STA IOB.TRACK
9499- A5 0A 2220 LDA THIS.TS.SECTOR
949B- 8D ED B7 2230 STA IOB.SECTOR
949E- 4C 21 94 2240 JMP WRITE.TS.LIST      write this list
2250 *-----
2260 *
2270 *       READ OR WRITE A SECTOR
2280 *
2290 *       READ.NEXT.SECTOR
94A1- A9 01 2300 LDA #1             read
94A3- 2C 2310 .HS 2C          skip 2
2320 *       WRITE.NEXT.SECTOR
94A4- A9 02 2330 LDA #2             write
94A6- 8D F4 B7 2340 STA IOB.OPCODE
94A9- 18 2350 CLC
94AA- 20 C6 94 2360 JSR NEXT.TS.ENTRY  get track/sector
94AD- D0 03 2370 BNE .1             .EQ. means EOF
94AF- E6 02 2380 INC EOF.FLAG      raise the flag
94B1- 60 2390 RTS             and quit
2400
94B2- 8D EC B7 2410 .1 STA IOB.TRACK
94B5- 8C ED B7 2420 STY IOB.SECTOR
2430
2440 *       RWTS.CALLER
94B8- 9C EB B7 2450 STZ IOB.VOLUME
94BB- 20 E3 03 2460 JSR GET.IOB
94BE- 20 D9 03 2470 JSR CALL.RWTS
94C1- 90 02 2480 BCC .1             .CS. if error
94C3- 85 03 2490 STA ERROR.FLAG  pass code back

```


Serial Pro® The intelligent multifunction card from Applied Engineering.



Serial Pro is a powerful multifunction card for the Apple IIe, II+, IIgs and compatible computers. Serial Pro combines a powerful serial port for interfacing to a printer, modem or other serial device, with a full function clock/calendar. Serial Pro's serial port is compatible with virtually all letter quality and dot matrix printers, including Apple's Imagewriter and Imagewriter II, and can communicate with the fastest external modems at baud rates from 50 to 19,200.

Set up is easy, because unlike other cards, there are no switches to program serial parameters or disks to use. Serial Pro has a built-in setup screen that allows all parameters to be set. And on-screen help that makes it super easy: An on-board battery stores your selections for over 20 years, but you can change them at any time. Serial Pro is the perfect upgrade for Super Serial card owners looking for more performance or a built-in clock.

Serial Port Features

- Fully Super Serial Card compatible but more versatile and easier to use
- Compatible with printers
- Compatible with modems
- Prints graphics to printer (without software) in normal mode, 90° rotation, double size, side by side, and black/white inversion
- Variable word length, number of stop bits and parity selection
- Crystal controlled circuits are used to provide ultra accurate baud rates

Clock Calendar Features

- ProDOS, DOS and AppleWorks compatible. Displays time and date on the AppleWorks screen
- Automatically time and date stamps AppleWorks files
- 20 year auto recharging nickel-cadmium battery
- Realtime interrupts
- Built-in on screen time and date settings (no software needed)
- Does not use phantom slots

Serial Pro is perfect for Apple IIe, II+, IIgs or compatible owners in need of a serial port or clock/calendar or both. But unlike other multifunction cards on the market, Serial Pro does not use phantom slots, so all the slots in your computer are useable.

Serial Pro comes complete with manual, cable and instructions for connecting to all the leading printers and modems.
Serial Pro \$159

Order Today! See your dealer or call Applied Engineering, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

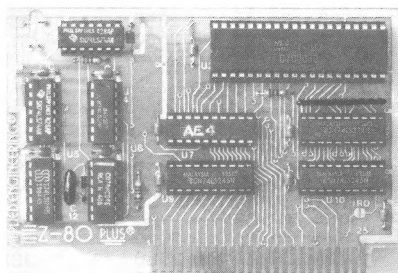
AE APPLIED ENGINEERING™

The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

With the Z-80 Plus®, run over 5000 new CP/M® programs.



*Now, get two computers
in one, and all the
advantages of both.*

Enter the CP/M world with the new Z-80 Plus card from Applied Engineering, and introduce your Apple IIe, IIgs or II+ to thousands of CP/M programs. Only the Z-80 Plus comes standard with the new 5.1 software, the most advanced system ever for running CP/M programs.

The new 5.1 boasts advanced features like built-in disk emulation for popular memory expansion boards, boosting both system speed and storage capacity. And menu-driven utilities that let you get to work faster. The Z-80 Plus also lets you run older CP/M programs—all the way down to Version 1.6 (2.2 is the most popular).

The Z-80 Plus is the only card on the market capable of accessing more than 64K. If you have an extended 80 column card, all 128K is usable. And if you have RamWorks, RamFactor, osRam, osRam Plus, or an Apple memory card in your IIe, IIgs or II+, up to 1088K is available.

Each Z-80 Plus comes with our CP/M Ram Drive software, enabling most memory cards to work as a high-speed Ram disk so CP/M software can run up to *twenty times faster*. So packages like WordStar, dBASE II, Turbo Pascal and most others run at blinding speed.

Simply plug the Z-80 Plus into any slot in your Apple. You'll get the benefits of two computers in one—all at an unbelievably low price (only \$149!).

- Fully compatible with ALL CP/M software
- Fully compatible with 5¼" and 3½" disk drives as well as most hard disks including the Sider
- Fully compatible with Microsoft disks (no pre-boot required)
- Specifically designed for high speed operation
- Runs WordStar, dBASE II, Turbo Pascal, and ALL other CP/M software with *no pre-boot*
- Semi-custom I.C. and low parts count allows Z-80 Plus to fly through CP/M programs with extremely low power consumption (we use the Z-80B)
- Does EVERYTHING other Z-80 boards do, *plus* Z-80 interrupts
- Five-year warranty

To get your Z-80 Plus or for more information see your dealer, or call 214-241-6060 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

CP/M is a registered trademark of Digital Research.

```

94C5- 60      2500 .1      RTS
                *-----*
                2510 *
                2520 *
                2530 *      RETURN NEXT TRACK/SECTOR
                2540 *
                2550 NEXT.TS.ENTRY
94C6- A6 06      2560 LDX TS.INDEX
94C8- F0 0B      2570 BEQ .2      done with this list
94CA- E6 06      2580 INC TS.INDEX      bump index
94CC- E6 06      2590 INC TS.INDEX      for next time
94CE- BC A7 9B   2600 LDY TS.BUFFER+1,X
94D1- BD A6 9B   2610 LDA TS.BUFFER,X      return .EQ. if eof
94D4- 60         2620 .1      RTS
                2630
94D5- AD A7 9B   2640 .2      LDA TS.BUFFER+1      track of next t/s list
94D8- F0 16      2650 BEQ .3      no next list
94DA- 8D EC B7   2660 STA IOB.TRACK
94DD- AD A8 9B   2670 LDA TS.BUFFER+2      sector
94E0- 8D ED B7   2680 STA IOB.SECTOR
94E3- AD F4 B7   2690 LDA IOB.OPCODE      read or write
94E6- 48         2700 PHA      stash it
94E7- 20 1E 94   2710 JSR READ.TS.LIST      read next one
94EA- 68         2720 PLA
94EB- 8D F4 B7   2730 STA IOB.OPCODE      restore
94EE- 80 D6      2740 BRA NEXT.TS.ENTRY    and go on
                2750
94F0- A5 0B      2760 .3      LDA ALLOCATE.FLAG      are we allocating?
94F2- F0 E0      2770 BEQ .1      no, it's eof time
94F4- 20 44 B2   2780 JSR ALLOCATE.SECTOR for next list
94F7- 8D A8 9B   2790 STA TS.BUFFER+2      sector
94FA- AD F1 B5   2800 LDA CURRENT.TRACK
94FD- 8D A7 9B   2810 STA TS.BUFFER+1      and track
9500- A5 09      2820 LDA THIS.TS.TRACK
9502- 8D EC B7   2830 STA IOB.TRACK
9505- A5 0A      2840 LDA THIS.TS.SECTOR
9507- 8D ED B7   2850 STA IOB.SECTOR
950A- 20 21 94   2860 JSR WRITE.TS.LIST      write full sector
                2870
950D- AD A7 9B   2880 LDA TS.BUFFER+1
9510- 85 09      2890 STA THIS.TS.TRACK      keep track
9512- AD A8 9B   2900 LDA TS.BUFFER+2
9515- 85 0A      2910 STA THIS.TS.SECTOR
9517- A2 00      2920 LDX #0
9519- 9E A6 9B   2930 .4      STZ TS.BUFFER,X      clear the buffer
951C- E8         2940 INX
951D- D0 FA      2950 BNE .4
951F- 18         2960 CLC
9520- A5 07      2970 LDA TS.OFFSET
9522- 69 7A      2980 ADC #122      update offset
9524- 85 07      2990 STA TS.OFFSET
9526- 8D AB 9B   3000 STA TS.BUFFER+5
9529- A5 08      3010 LDA TS.OFFSET+1
952B- 69 00      3020 ADC #0
952D- 85 08      3030 STA TS.OFFSET+1
952F- 8D AC 9B   3040 STA TS.BUFFER+6
9532- E6 04      3050 INC PGM.LENGTH      count new list sector
9534- D0 02      3060 BNE .5
9536- E6 05      3070 INC PGM.LENGTH+1
9538- 80 8C      3080 .5      BRA NEXT.TS.ENTRY    and go on
                3090 *-----*
                3100 FILE.MANAGER.ERROR
953A- 20 43 95   3110 JSR REMOVE.DOS.PATCHES clean house
953D- AD C5 B5   3120 LDA FM.ERROR
9540- 85 03      3130 STA ERROR.FLAG      and report error
9542- 60         3140 RTS
                3150 *-----*
                3160 *
                3170 *      HANDLE TWO PATCHES TO FILE MANAGER
                3180 *
                3190 REMOVE.DOS.PATCHES
9543- A2 00      3200 LDX #0
9545- 2C         3210 .HS 2C      skip two bytes
                3220 INSTALL.DOS.PATCHES
9546- A2 04      3230 LDX #4
9548- BD 61 95   3240 LDA DOS.PATCHES,X

```

```

954B- 8D FD AF 3250      STA $AFFD
954E- BD 62 95 3260      LDA DOS.PATCHES+1,X
9551- 8D 92 B3 3270      STA $B392
9554- BD 63 95 3280      LDA DOS.PATCHES+2,X
9557- 8D 93 B3 3290      STA $B393
955A- BD 64 95 3300      LDA DOS.PATCHES+3,X
955D- 8D 94 B3 3310      STA $B394
9560- 60                RTS
3320
3330 *-----
3340 DOS.PATCHES
3350 * Original Code
9561- AC                .HS AC      LDY opcode
9562- AE 9B B3 3370      LDX $B39B
3380 * Patching Code
9565- 60                RTS          don't allow sector allocate
3400 *                  routine to read/write/VTOC
9566- 4C 3A 95 3410      JMP FILE.MANAGER.ERROR trap error exits,
3420 *                  because we didn't really call
3430 *                  File Manager legitimately.
3440 *-----

```

EnterSoft

Basic-like macros for the S-C Macro Assembler.

Available for the Apple II, II+, IIe, and IIC.

Uses whatever DOS your S-C Macro Assembler uses.

Soon to be released for PRODOS.

Source code included with every disk!!!

Cost? A paltry \$30.00!!!

To Order - Send Check or Money Order to:



Mark Manning
c/o Baggy Games
P. O. Box 591894
Houston, Texas 77259-1894

Apple is a registered Trademark of Apple Computer Inc.

The S-C Macro Assembler is produced by The S-C Software Corporation.

EnterSoft is produced by Mark Manning/Simulacron I/Baggy Games

A Smarter Hexadecimal Memory Search.....Bob Sander-Cederlof

Today Rick Hayner called, wanting some help in locating places inside Rak-Ware's DISASM where it calls SETVID and various other subroutines in the monitor. Rick is blind, so he uses the ECHO speech synthesizer to operate his Apple. Programs like DISASM keep un-hooking the speech synthesizer, so he has to patch it until it keeps talking. It takes a long time to search through a whole program for all the possible kinds of code which could un-hook the output device.

We used the monitor "S" command, which can search for a single byte or a two-byte value. This command is not in every Apple monitor, but it was in the one in my computer. We found most of the places, but not all. The trouble was we didn't know exactly what addresses we were looking for, because there are so many possibilities.

Anyway, I kept thinking we needed some kind of searching program which could look through DISASM or whatever more intelligently. What an idea! Let the computer work for us! I came up with SMART.SEARCH, which fills the bill.

SMART.SEARCH can search through a range of memory for any string of bytes up to 255 bytes long. (The listing is prettier if the string length is less than 80, though.) Three strings supplied by the caller define the content of matching strings. The first string is a "mask". Each byte of memory will be ANDed with the corresponding byte in the MASK string before being compared with the key. This lets you ignore certain bits in certain bytes during the search (such as the high bit in ASCII bytes). The key is defined by two strings, providing a lower and an upper limit for values in each byte.

For example, to search for JSR's to subroutines in the monitor, I use a mask string of "FF 00 FF", a lower limit of "20 00 F8", and an upper limit of "20 00 FF". Notice the middle byte always matches, so we trigger on any JSR's to any address between \$F800 and \$FFFF.

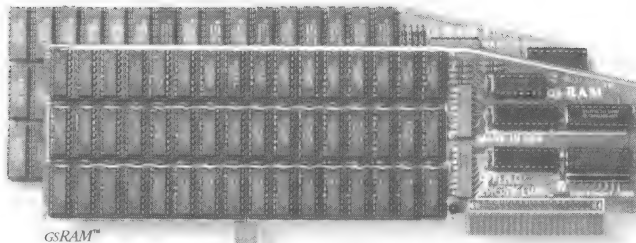
In the listing which follows, Lines 1180-1420 are a demonstration driver for the SMART.SEARCH subroutine. Each call to SMART.SEARCH searches the range from \$0800 to \$14FF, which happens to be where DISASM loads. I first printed out all the monitor calls, then all the JMPs into the monitor, and then all STA, STX, and STY instructions which store into any address with the low-order byte equal to \$36 through \$39 (the I/O hooks are at \$0036-0039). We might want to also search for JMPs and JSRs to \$03EA, which is the DOS I/O hook subroutine, and for stores into the DOS hook area starting at \$AA53, but that is "left as an exercise for the reader".

When SMART.SEARCH finds a matching string, it prints out the address of the first byte, and all the bytes of the matching string. SMART.SEARCH displays as many columns of addresses and strings as in can on an 80-column line. If you want a different line length, change the value in line 2050.

Insist on **gsRAM™** When You Buy Your **IIgs™**

*Expand the IIgs RAM and ROM with the **gsRAM** or **gsRAM Plus**
with ROM Pak. Available now with 256K to 8 MEG!*

gsRAM Plus™



gsRAM™

Remember the 16K cards for the II+ and the 64K cards for the IIe? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

gsRAM and gsRAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only gsRAM and gsRAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in Hi-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

gsRAM for More AppleWorks Power

Only gsRAM and gsRAM Plus eliminate AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000 and only gsRAM and gsRAM Plus increase the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only gsRAM and gsRAM Plus offer a built-in printer buffer so you can continue using AppleWorks while your printer is printing. gsRAM and gsRAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can

even have Pinpoint or Macroworks and your favorite spelling checker in RAM for instant response. gsRAM and gsRAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIgs into a Giant

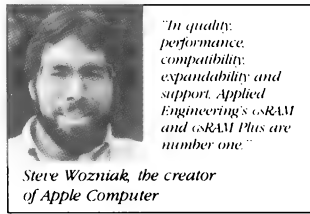
Simply plug gsRAM into the IIgs memory expansion slot and you've got up to 8 megabytes of RAM at your fingertips—all of it instantly and automatically recognized by the IIgs. gsRAM is compatible with all IIgs software, including AppleWorks, as well as BASIC®, ProDOS, DOS 3.3, PASCAL®, "C" and CP/M®.

Grow by Bytes or Megabytes

We offer gsRAM in two configurations so you can increase your memory 256K at a time (gsRAM) or a megabyte at a time (gsRAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded gsRAM operates at only 375 ma, and gsRAM Plus at only 270 ma (even with 6 megabytes on board!).

gsRAM—for Normal Memory Requirements

gsRAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a gsRAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to gsRAM Plus for a small charge.



"In quality, performance, compatibility, expandability and support, Applied Engineering's gsRAM and gsRAM Plus are number one."

*Steve Wozniak, the creator
of Apple Computer*

With an optional piggyback card, you can expand gsRAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

gsRAM Plus—for Growing by Leaps and Bounds

gsRAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a gsRAM Plus with less memory and easily expand it in the future.

gsRAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both gsRAM and gsRAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your gsRAM or gsRAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIgs's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We invented the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure gsRAM and gsRAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

gsRAM with 256K	\$169
gsRAM with 512K	\$219
gsRAM with 1 MEG	\$299
gsRAM with 1.5 MEG	\$379
gsRAM with 2-8 MEG	CALL
gsRAM PLUS with 1 MEG	\$459
gsRAM PLUS with 2 MEG	\$759
gsRAM PLUS with 3-8 MEG	CALL

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6.4% sales tax. Add \$10.00 outside U.S.A.

AE APPLIED ENGINEERING™
The Apple enhancement experts.
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research, Inc.

SMART.SEARCH has to be driven by a series of calls, which are assembled together with the SMART.SEARCH subroutine. Before you assemble, change line 1170 to set the origin to a place that does not conflict with the memory you are going to examine. Calls to SMART.SEARCH require a lot of information. I decided to use the technique of including all that information immediately after the JSR SMART.SEARCH. You can see how to code the information by studying lines 1180-1420.

Inside SMART.SEARCH, there are three main sections. The first gets information from the caller and sets up a lot of addresses in page-zero vectors. The second searches for a string that matches. The third prints a matching address and string.

Lines 1450-1740, along with several subroutines in lines 2350-2560, pull in all the caller's information. After the information has all been set up, RETURN contains the address SMART.SEARCH to which should go when it is finished.

Lines 1750-1860 compare for a matching string. Lines 1870-2090 print the matching address and string if one is found. Lines 2110-2230 increment the search address for the next comparison. If not past the end address, this code will branch back to do another comparison.

Lines 2110-2120 and 2250-2330 allow the output to be paused and restarted, or aborted by pressing any key. The RETURN key will terminate the particular SMART.SEARCH in progress.

While writing SMART.SEARCH, I thought of many other possibilities. You might like to try some of them. You could adapt it to search through all the sectors of a floppy disk. You could modify the comparison algorithm to include more options. You could search through all the bytes on an extended memory card (such as RAMWORKS or RAMFACTOR). You could modify the output format. I thought of including a FORMAT string, that could specify both byte order and whether to print the bytes found in hex or in ASCII. Or, you could just use the "shell" which handles the parameter passage, and code an entirely different function. If you have an Apple IIgs, you might want to modify SMART.SEARCH so that it looks like a "Tool", and have it installed at boot-up time.

If you come up with some great ones, let us have a look. We might like to pass them on to the rest of you!

```

                                1000 #SAVE SMART.SEARCH
                                1010 #-----
00-                               1020 PNTR .EQ $00,01
02-                               1030 PEND .EQ $02,03
04-                               1040 PMASK .EQ $04,05
06-                               1050 PLOWER .EQ $06,07

```

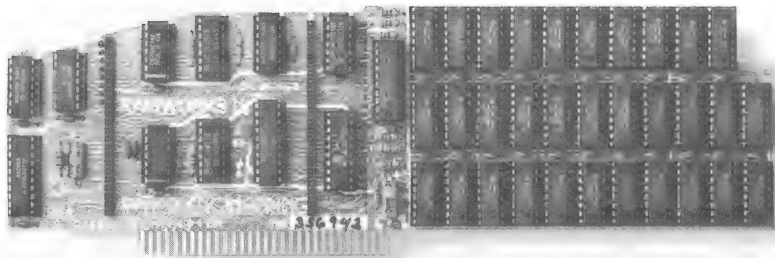
```

08-          1060 PUPPER .EQ $08,09
0A-          1070 LENGTH .EQ $0A
0B-          1080 FWIDTH .EQ $0B
0C-          1090 COLUMN .EQ $0C
0D-          1100 RETURN .EQ $0D,0E
          1110 *-----
F941-        1120 PRNTAX .EQ $F941
FDED-        1130 COUT .EQ $FDED
FD8E-        1140 CROUT .EQ $FD8E
FDDA-        1150 PRBYTE .EQ $FDDA
          1160 *-----
          1170 .OR $3000
          1180 T
3000- 20 31 30 1190 JSR SMART.SEARCH
3003- 00 08 1200 .DA $0800 Starting Address
3005- 00 15 1210 .DA $1500 Ending address + 1
3007- 03 1220 .HS 03 Length of strings
3008- FF 00 FF 1230 .HS FF.00.FF Mask string
300B- 20 00 F8 1240 .HS 20.00.F8 Lower limit string
300E- 20 00 FF 1250 .HS 20.00.FF Upper limit string
          1260 *-----
3011- 20 31 30 1270 JSR SMART.SEARCH
3014- 00 08 1280 .DA $0800 Starting Address
3016- 00 15 1290 .DA $1500 Ending address + 1
3018- 03 1300 .HS 03 Length of strings
3019- FF 00 FF 1310 .HS FF.00.FF Mask string
301C- 4C 00 F8 1320 .HS 4C.00.F8 Lower limit string
301F- 4C 00 FF 1330 .HS 4C.00.FF Upper limit string
          1340 *-----
3022- 20 31 30 1350 JSR SMART.SEARCH
3025- 00 08 1360 .DA $0800 Starting Address
3027- 00 15 1370 .DA $1500 Ending address + 1
3029- 02 1380 .HS 02 Length of strings
302A- FF FF 1390 .HS FF.FF Mask string
302C- 81 36 1400 .HS 81.36 Lower limit string
302E- 9D 39 1410 .HS 9D.39 Upper limit string
3030- 60 1420 RTS
          1430 *-----
          1440 SMART.SEARCH
3031- 20 8E FD 1450 JSR CROUT
3034- 68 1460 PLA GET RETURN ADDRESS
3035- 85 OD 1470 STA RETURN
3037- 68 1480 PLA
3038- 85 OE 1490 STA RETURN+1
          1500 *---Get Parameters after JSR---
303A- 20 E5 30 1510 JSR GET.WORD Get Starting Address
303D- 86 00 1520 STX PNTR
303F- 85 01 1530 STA PNTR+1
3041- 20 E5 30 1540 JSR GET.WORD Get Ending Address + 1
3044- 86 02 1550 STX PEND
3046- 85 03 1560 STA PEND+1
3048- 20 E9 30 1570 JSR GET.BYTE Get Search String Length
304B- 85 0A 1580 STA LENGTH
304D- 0A 1590 ASL Calc Field Width
304E- 69 07 1600 ADC #7
3050- 85 0B 1610 STA FWIDTH
3052- 85 0C 1620 STA COLUMN Start column counter
3054- 20 E9 30 1630 JSR GET.BYTE Point at Mask String
3057- A5 OD 1640 LDA RETURN
3059- 85 04 1650 STA PMASK
305B- A5 OE 1660 LDA RETURN+1
305D- 85 05 1670 STA PMASK+1
305F- 20 F4 30 1680 JSR ADD.LENGTH.TO.RETURN
3062- 86 06 1690 STX PLOWER Point at Lower Limit String
3064- 85 07 1700 STA PLOWER+1
3066- 20 F4 30 1710 JSR ADD.LENGTH.TO.RETURN
3069- 86 08 1720 STX PUPPER Point at Upper Limit String
306B- 85 09 1730 STA PUPPER+1
306D- 20 F4 30 1740 JSR ADD.LENGTH.TO.RETURN
          1750 *---Compare strings-----
3070- A0 00 1760 LDY #0
3072- B1 00 1770 .2 LDA (PNTR),Y
3074- 31 04 1780 AND (PMASK),Y
3076- D1 06 1790 CMP (PLOWER),Y
3078- 90 3F 1800 BCC .6

```

RamWorks® III

Patented Performance from the Recognized Leader



With battery backed RAM port, RGB port, increased memory capacity, full software compatibility and more compact design, RamWorks III is a generation ahead.

RamWorks III is the newest 3rd generation RAM card for the Apple IIe. It incorporates all of the technology and improvements that years of experience and over a hundred thousand sales have given us. By selling more memory cards than anyone else and listening to our customers, we were able to design a memory card that has the ultimate in performance, quality, compatibility and ease of use. A design so advanced it's patented. We call it RamWorks III, you'll call it awesome!

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K, and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 256K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

Only RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 25,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 15,000. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your printer to stop before returning to AppleWorks. RamWorks even expands the clipboard. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, *nothing* comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, HowardSoft, FlashCalc, Pro-File, Managing Your Money, SuperCalc 3a, and MagiCalc to name a few (and *all* hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601081) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all

Apple IIe's, enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 meg is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 2 MEG, or 16 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector. You can also choose non-volatile, power independent expanders allowing permanent storage for up to 20 years.

It Even Corrects Mistakes.

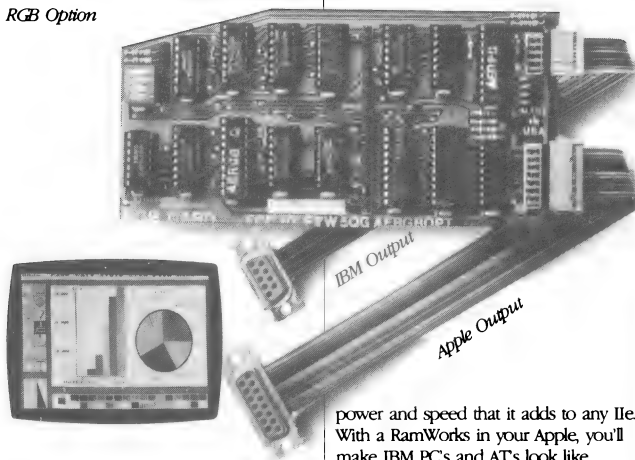
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB

Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both Apple standard and IBM standard RGB outputs (cables included). The RGB output plugs into the back of RamWorks with no slot 1 inter-

RGB Option



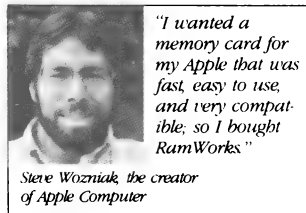
ference and remember you can order the RGB option with your RamWorks or add it on at a later date.

True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 meg of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 yet maintains full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." inCider magazine said "RamWorks is the most



Steve Wozniak, the creator of Apple Computer

"I wanted a memory card for my Apple that was fast, easy to use, and very compatible, so I bought RamWorks."

powerful auxiliary slot memory card available for your IIe, and I rate it four stars...For my money, Applied Engineering's RamWorks is king of the hill."

Apple experts everywhere are impressed by RamWorks's expandability, versatility, ease of use, and the sheer

coverage no matter where you purchase

- Built-in super sharp 80 column display, (U.S. Patent #4601081)
- Expandable to 1 MEG on main card
- Expandable to 16 meg with expander cards, with NO slot 1 interference
- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- The only large RAM card that's 100% compatible with all IIe software
- RamDrive™ the ultimate disk emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible, RGB option featuring ultra high resolution color graphics and multiple text colors, with cables for both Apple and IBM type monitors
- Built-in self diagnostics software
- Lowest power consumption (U.S. Patent #4601081)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Wozniak and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$219
RamWorks III with 512K	\$269
RamWorks III with 1 MEG	\$369
RamWorks III with 1.5 MEG	\$539
RamWorks III with 2 to 16 MEG	CALL
65C816 16 Bit Card	\$159
RGB Option	\$129
Optional Software:	
Pinpoint with RAM	
Enhancement Software	\$79
VIP Professional	\$219

power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PC's and AT's look like slowpokes.

Quality and Support of the Industry Leader

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All

- 15 day money back guarantee
- 5 year hassle free warranty insures

AE Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

RamWorks III. The industry standard for memory expansion of the Apple IIe. ORDER YOUR RamWorks III TODAY. 9 a.m. to 11 p.m. 7 days, or send check or money order to Applied Engineering. MasterCard, Visa and C.O.D. welcome. Texas residents add 5½% sales tax. Add \$10.00 if outside U.S.A.

```

307A- D1 08      1810      CMP (PUPPER),Y
307C- 90 02      1820      BCC .3      ...YES
307E- D0 39      1830      BNE .6
3080- C8          1840      .3      INY
3081- C4 0A      1850      CPY LENGTH
3083- 90 ED      1860      BCC .2      ...MORE TO COMPARE
3085- A5 01      1870      *---Found a matching string-----
3087- A6 00      1880      LDA PNTR+1      Print the Address
3089- 20 41      1890      LDX PNTR
3089- 20 41      1900      JSR PRNTAX
308C- A9 AD      1910      LDA #"- "
308E- 20 ED      1920      JSR COUT
3091- A0 00      1930      LDY #0      Print the String
3093- B1 00      1940      .4      LDA (PNTR),Y
3095- 20 DA      1950      JSR PRBYTE
3098- C8          1960      INY
3099- C4 0A      1970      CPY LENGTH
309B- 90 F6      1980      BCC .4
309D- A9 A0      1990      LDA # "      Tab to next column
309F- 20 ED      2000      JSR COUT
30A2- 20 ED      2010      JSR COUT
30A5- A5 0B      2020      *---Check if end of line-----
30A7- 65 0C      2030      LDA FWIDTH
30A9- C9 50      2040      ADC COLUMN
30AB- 90 05      2050      CMP #80
30AD- 20 8E      2060      BCC .5
30AD- 20 8E      2070      JSR CROUT
30B0- A5 0B      2080      LDA FWIDTH
30B2- 85 0C      2090      STA COLUMN
30B4- AD 00      2100      .5      *---Check for pause/abort-----
30B7- 30 16      2110      LDA $C000
30B7- 30 16      2120      BMI .7      PAUSE OR ABORT
30B9- E6 00      2130      *---Advance Pointer-----
30BB- D0 B3      2140      .6      INC PNTR
30BD- E6 01      2150      BNE .1
30BF- A5 00      2160      INC PNTR+1
30C1- C5 02      2170      LDA PNTR
30C3- A5 01      2180      CMP PEND
30C5- E5 03      2190      LDA PNTR+1
30C7- 90 A7      2200      SBC PEND+1
30C9- 20 8E      2210      BCC .1
30C9- 20 8E      2220      JSR CROUT
30CC- 6C 0D      2230      JMP (RETURN)
30CF- 8D 10      2240      *-----
30D2- C9 8D      2250      .7      STA $C010
30D4- F0 0C      2260      CMP #$8D
30D6- AD 00      2270      BEQ .9      ...ABORT
30D9- 10 FB      2280      .8      LDA $C000
30DB- 8D 10      2290      BPL .8      ...PAUSE
30DE- C9 8D      2300      STA $C010
30E0- D0 D7      2310      CMP #$8D
30E2- 6C 0D      2320      BNE .6      ...END OF PAUSE
30E2- 6C 0D      2330      .9      JMP (RETURN)
30E5- 20 E9      2340      *-----
30E8- AA          2350      GET.WORD
30E8- AA          2360      JSR GET.BYTE
30E9- E6 0D      2370      TAX
30EB- D0 02      2380      GET.BYTE
30ED- E6 0E      2390      INC RETURN
30EF- A0 00      2400      BNE .1
30F1- B1 0D      2410      INC RETURN+1
30F3- 60          2420      .1      LDY #0
30F4- 18          2430      LDA (RETURN),Y
30F5- A5 0D      2440      RTS
30F7- 65 0A      2450      *-----
30F9- 85 0D      2460      ADD.LENGTH.TO.RETURN
30FB- AA          2470      CLC
30FC- A5 0E      2480      LDA RETURN
30FE- 69 00      2490      ADC LENGTH
3100- 85 0E      2500      STA RETURN
3102- 60          2510      TAX
3102- 60          2520      LDA RETURN+1
3102- 60          2530      ADC #0
3102- 60          2540      STA RETURN+1
3102- 60          2550      RTS

```

For the last year or so Doug McComsey has been fine tuning this program, and the result is a tool which I like, use, and recommend.

ProVIEW is essentially a slick ZAP program for ProDOS disks and memory. On an Apple //c, //e, or //gs in 80-column mode ProVIEW gives you a window into RAM, ROM, ProDOS files, and ProDOS disk blocks. You can examine, modify, and update any portion of a file or disk block. ProVIEW does not recognize DOS disks, and will not work in a II, IIPlus, IIclone, or anything in 40-column mode. Nevertheless, these restrictions are more than overcome by its features and ease-of-use.

The various control and display screens of ProVIEW look and function a lot like Appleworks. As a result, you very quickly outgrow the need for a reference manual. All of the keystrokes you need to use are natural and easy to remember; help is only a keystroke away at all times, using the Apple-? combination.

ProVIEW executes without disturbing the S-C Macro Assembler or BASIC.SYSTEM. When you want to use it, you simply insert the disk and type "-PV". Or, if you have it on your hard disk or RAM-disk you can get it even quicker. When you are through, a simple ESCAPE keystroke gets you back to the assembler or to Applesoft.

Doug has asked us to handle sales of ProVIEW. Our introductory price is only \$20. You will get documentation and a copyable disk, with one of the handiest tools you've ever purchased.

RAMWORKS™

**ACCEPT NO SUBSTITUTES.
BECAUSE THERE AREN'T ANY.**

There's only one card like RamWorks. We've got the best hardware design. We supply the best software and we've got the best support from software companies.

If someone tempts you with an imitation, please get both sides of the story. You'll discover why RamWorks offers the best enhancements to AppleWorks and other programs, and at the lowest price.

GUARANTEED!

**214-241-6060
9 AM - 11 PM**


**"We Set the Standard"
APPLIED ENGINEERING**

DOS Anatomy

AN UNPRECEDENTED DISASSEMBLY

FEATURES:

- ONLY COMPLETE DISASSEMBLY of the DOS 3.3 operating system.
- entire operating system is described in BOTH LINEAR AND FORMATTED FORMS.
- all disassemblies are HEAVILY COMMENTED.
- describes numerous tips and techniques for modifying DOS and implementing sophisticated PROTECTION SCHEMES.
- describes a LEGAL SOURCE OF FREE COMPUTER PAPER that may be located in your area.
- Includes a UNIQUE TEXT-FILE READING PROGRAM (called TEXT READER) that can be used to read both random-access and sequential text files at binary speeds. This machine-language program is LOADED WITH OPTIONS and can be configured to accommodate various printers and interface cards. (Source code for the TEXT READER program is available at an additional cost).
- entire software package is UNPROTECTED AND GUARANTEED.
- high quality / low price: 64 FILES ON 15 DOUBLE-SIDED DISKS
FOR ONLY \$100 (US Funds).

DESIGNED FOR:

- software developers - computer magazine authors - protection artists
- computer hackers - instructors of assembly language
- any assembly language programmer who frequently foregoes sleep in an effort to reach new plateaus of understanding.

SYSTEM REQUIREMENTS:

- 1) Apple II+ (with 16K ram card), Apple IIe or Apple IIc.
 - 2) One disk drive (using 5-1/4 inch diskettes).
 - 3) DOS 3.3 operating system. (Note: In order to conserve disk space, all files are housed on DOS-less data disks. Therefore, you must boot with a normal version of DOS 3.3 before this software package can be used.)
 - 4) Printer that can support a 132-character line length in condensed or normal modes. (A smaller printer width can be used but is not recommended.)
- Optional---
- 5) MERLIN (a.k.a. "BIG MAC") assembler. (An assembler is not required if you just wish to review the text file contents. Text files containing numerical and alphabetical cross-referenced symbol tables are provided to assist programmers who own a different brand of assembler.)
 - 6) The book "BENEATH APPLE DOS" would prove helpful.

To order, send cheque or money order for \$100 (US Funds) to:

WINDSOR SCIENTIFIC LTD.
3621 - 109 Avenue Edmonton Alberta
Canada T5W 0G6
Phone (403) 479-2779

DOS 3.3, MERLIN and BENEATH APPLE DOS are respective trademarks or products of Apple Computer Inc.,
Roger Wagner Publishing Inc., and Quality Software.

Display the IIgs Tool Tables.....Bob Sander-Cederlof

While exploring in my IIgs, a map is handy. There is a lot of RAM and a lot of ROM, and it is practically impossible to keep everything in my head. It is also ridiculous to try drawing such a map with pencil and paper. Why not use the computer to document itself?

The tools in the IIgs are all accessed through a hierarchy of tables. All you need to know is the address of the first (top) table. The top table starts with a 4-byte value which is the number of secondary tables. Following this count are a list of 4-byte addresses which point to each of the secondary tables. Each secondary table begins with a 4-byte count of the number of tools in that toolset, and continues with a list of 4-byte addresses pointing to the code for each tool.

These tables may be either in RAM or ROM. When you turn on your IIgs certain values are loaded into bank \$E1. Among these is a pointer at \$E1/03C0 which contains a 3-byte address for the top tool table. In my IIgs this is at \$FE/013F. All of the secondary tables are also in bank \$FE of ROM. However, after booting the system disk most of the tables have been rebuilt in RAM. About half of the toolsets are not resident in ROM anyway, so those tables are meaningless until the system disk has been booted and the RAM-based toolsets loaded. Many of the ROM-based toolsets are already somewhat obsolete, because the system disk builds RAM-based tables and loads modified tool code into RAM.

The following program will start at \$E1/03C0 and follow the hierarchy of tables. The addresses for all of the tools in every toolset will be displayed. If your printer is on and connected, the tables will be listed on the printer as well.

Although my program runs entirely in Emulation Mode, it still uses some of the new addressing modes of the 65816. Line 1170 shows an example of long indexed mode, and is part of a loop copying the address of the top table into a page-zero pointer. Line 1390 is one of many examples using the long-indirect-indexed mode. In this mode a 3-byte address in page zero is added to the current value in the Y-register to get the effective address.

In the S-C Macro Assembler this long form is distinguished from the normal 2-byte indirect-indexed form by using a ">" before the operand: ">(zp),Y" for long-indirect-indexed, and "(zp),Y" for regular indirect-indexed. The "greater than" sign indicates mnemonically that the address at "zp" is three bytes long rather than two. The assemblers which followed the lead of ORCA/M use a different syntax: "[zp],Y" for long-indirect-indexed. I chose not to use the square brackets because the earlier Apple keyboards could not generate a left bracket ([), and because the right bracket was already in use for macro parameters. Future versions of S-C Macro may allow the ORCA syntax as well as the current form.

If you have a //gs, you will benefit by entering this program and running it. Once you know where the tools are located, you can dig in with the disassembler and find out what makes them tick!

```

1020      .OP 65816
1030      *-----
1040      *   This program runs in Emulation Mode
1050      *-----
FD8E-    1060 CROUT .EQ $FD8E
FDDA-    1070 PRBYTE .EQ $FDDA
FDED-    1080 COUT .EQ $FDED
1090      *-----
00-      1100 PTRB .EQ $00,01,02
03-      1110 PTRB .EQ $03,04,05
06-      1120 TOOL.NUMBER .EQ $06
07-      1130 TOOLSET.NO .EQ $07
1140      *-----
1150 LIST.TOOL.TABLES
000800-  A2 02      1160 LDX #2
000802-  BF C0 03 E1 1170 .1 LDA $E103C0,X      Start with main table
000806-  95 00      1180 STA PTRB,X          Address of table of toolset addresses
000808-  95 03      1190 STA PTRB,X
00080A-  CA        1200 DEX                3 bytes per address
00080B-  10 F5      1210 BPL .1
1220      *---Display main table-----
00080D-  A0 18      1230 LDY #Q.TOOLSET      Title for main table
00080F-  20 89 08   1240 JSR QTO
000812-  20 47 08   1250 JSR DISPLAY.A.TABLE  Addresses of toolset tables
1260      *---Loop for each toolset-----
000815-  A9 01      1270 LDA #1              Start with toolset #1
000817-  85 07      1280 STA TOOLSET.NO
000819-  A0 3A      1290 .2 LDY #Q.TOOL      Toolset title
00081B-  20 89 08   1300 JSR QTO
00081E-  A5 07      1310 LDA TOOLSET.NO      Print the # too
000820-  20 DA FD   1320 JSR PRBYTE
000823-  A0 28      1330 LDY #Q.LINE          Underline it
000825-  20 89 08   1340 JSR QTO
000828-  A5 07      1350 LDA TOOLSET.NO      Make toolset# into index
00082A-  0A        1360 ASL                by multiplying by 4
00082B-  0A        1370 ASL
00082C-  A8        1380 TAY
00082D-  B7 03      1390 LDA >(PTRB),Y      Get address of toolset table
00082F-  85 00      1400 STA PTRB            lo-byte
000831-  C8        1410 INY
000832-  B7 03      1420 LDA >(PTRB),Y
000834-  85 01      1430 STA PTRB+1          mid-byte
000836-  C8        1440 INY
000837-  B7 03      1450 LDA >(PTRB),Y
000839-  85 02      1460 STA PTRB+2          hi-byte
00083B-  20 47 08   1470 JSR DISPLAY.A.TABLE
00083E-  E6 07      1480 INC TOOLSET.NO      Next toolset...
000840-  A5 07      1490 LDA TOOLSET.NO
000842-  C7 03      1500 CMP >(PTRB)          Compare to number of existing sets
000844-  90 D3      1510 BCC .2              ...more toolsets
000846-  60        1520 RTS
1530      *-----
1540 DISPLAY.A.TABLE
000847-  A0 00      1550 LDY #Q.TITLE        Title for the table
000849-  20 89 08   1560 JSR QTO
00084C-  A9 01      1570 LDA #1
00084E-  85 06      1580 STA TOOL.NUMBER     Start with tool #1 in set
000850-  A0 10      1590 .1 LDY #Q.CRD        Newline and a dollar sign
000852-  20 89 08   1600 JSR QTO
000855-  A5 06      1610 LDA TOOL.NUMBER     Print the tool (function) # in hex
000857-  20 DA FD   1620 JSR PRBYTE
00085A-  A0 13      1630 LDY #Q.DOTS         Connect with dots to address column
00085C-  20 89 08   1640 JSR QTO
00085F-  A5 06      1650 LDA TOOL.NUMBER     Make tool# into an index by
000861-  0A        1660 ASL                multiplying by four
000862-  0A        1670 ASL
000863-  A8        1680 TAY
000864-  C8        1690 INY
000865-  C8        1700 INY
000866-  B7 00      1710 LDA >(PTRB),Y
000868-  20 DA FD   1720 JSR PRBYTE          Print hi-byte
00086B-  A9 AF      1730 LDA #"/"           Using bb/hhll format
00086D-  20 ED FD   1740 JSR COUT
000870-  88        1750 DEY                Print mid-byte
000871-  B7 00      1760 LDA >(PTRB),Y
000873-  20 DA FD   1770 JSR PRBYTE
000876-  88        1780 DEY                Print lo-byte
000877-  B7 00      1790 LDA >(PTRB),Y
000879-  20 DA FD   1800 JSR PRBYTE
00087C-  E6 06      1810 INC TOOL.NUMBER     Next tool # in set
00087E-  A5 06      1820 LDA TOOL.NUMBER
000880-  C7 00      1830 CMP >(PTRB)         Out of the set yet?
000882-  90 CC      1840 BCC .1              ...No, still more tools
000884-  60        1850 RTS

```

```

000885- 20 ED FD 1860 #-----
000888- C8 1870 QTO1 JSR COUT Print a string
000889- B9 8F 08 1880 INY
00088C- D0 F7 1890 QTO LDA QTS,Y Enter here with Y indexing message
00088E- 60 1900 BNE QTO1
1910 RTS
1920 #-----
1930 QTS
1940 Q.TITLE .EQ #-QTS
1950 .HS 8D

00-
00088F- 8D 1960 .AS -/Tool Address/
000890- D4 EF EF EC 1970 .HS 00
000894- A0 A0 A0 C1 1980 Q.CRD .EQ #-QTS
000898- E4 E4 F2 E5 1990 .HS 8D.A4.00
00089C- F3 F3 2000 Q.DOTS .EQ #-QTS
00089E- 00 2010 .AS -/...$/
10- 2020 .HS 00
00089F- 8D A4 00 2030 Q.TOOLSET .EQ #-QTS
13- 2040 .HS 8D
0008A2- AE AE AE A4
0008A6- 00
18-
0008A7- 8D
0008A8- A0 D4 EF EF
0008AC- EC F3 E5 F4
0008B0- A0 D4 E1 E2
0008B4- EC E5 F3
28-
0008B7- 8D
0008B8- AD AD AD AD
0008BC- AD AD AD AD
0008C0- AD AD AD AD
0008C4- AD AD AD AD
0008C8- 00 2080 .AS -/-----/
3A- 2090 .HS 00
0008C9- 8D 8D 8D 2100 Q.TOOL .EQ #-QTS
0008CC- D4 EF EF EC 2110 .HS 8D.8D.8D
0008D0- F3 A0 E9 EE
0008D4- A0 D3 E5 F4
0008D8- A0 A4
0008DA- 00 2120 .AS -/Tools in Set $/
2130 .HS 00

```

We Make Measurement And Control Easy!

12 BIT, 16 CHANNEL PROGRAMMABLE GAIN A/D

- All new 1984 design incorporates the latest in state-of-the-art IC technologies
- Complete 12 bit A/D converter, with an accuracy of 0.12%
- 16 single ended channels (single ended means that your signals are measured against the Apple's CHD3) or 8 differential channels. Must all the signals you will measure are single ended
- Software programmable full scale ranges, any of the 16 channels can have any range at any time. Under program control you can select any of the following ranges: ± 10 volts, ± 5 V, ± 2.5 V, ± 1.0 V, ± 500 mV, ± 250 mV, ± 100 mV, ± 50 mV, ± 25 mV
- Very fast conversion (25 micro seconds)
- Analog input resistance greater than 1,000,000 ohms
- Laser-trimmed scaling resistors
- Low power consumption through the use of CMOS devices
- The user connector has ± 12 and ± 12 volts on it so you can power your sensors
- Only elementary programming is required to use the A/D
- The entire system is on one standard size plug-in card that fits nicely inside the Apple
- System includes sample program on disk

PRICE \$319

A few applications may include monitoring of Φ flow \bullet temperature \bullet humidity \bullet wind speed \bullet wind direction \bullet light intensity \bullet pressure \bullet RPM \bullet soil moisture and many more

A/D & D/A

- Simple PC card
- 8 channels A/D
- 8 channels D/A
- Superfast conversion time
- Very easy programming
- Many analog ranges
- Manual contains sample applications

A/D SPECIFICATIONS

- 0.1% accuracy
- On-board memory
- Fast conversion (0.78 ms per channel)
- A/D process totally transparent to Apple (looks like memory)
- User programmable input ranges are 0 to 10 volts, 0 to 5, 5 to +5, -2.5 to +2.5, 5 to 0, 10 to 0

The A/D process takes place on a continuous channel sequencing basis. Data is automatically transferred to its proper location in the on-board RAM. No A/D converter could be easier to use

D/A SPECIFICATIONS

- 0.1% accuracy
 - On-board memory
 - On-board output buffer amplifiers can drive 5 MA
 - D/A process is totally transparent to the Apple (just poke the data)
 - Fast conversion (0.01 MS per channel)
 - User programmable output ranges are 0 to 5 volts and 0 to 10 volts
- The D/A section contains 8 digital to analog converters, with output buffer amplifiers and all interface logic on a single card. On-board latches are provided for each of the eight D/A converters. No D/A converter could be easier to use. The on-board amplifiers are laser trimmed during manufacture, thereby eliminating any requirement for oil setting.

PRICE \$199

SIGNAL CONDITIONER

Our 8 channel signal conditioner is designed for use with our A/D converters. This board incorporates 8 ± 1 E \pm mV inputs, which allow almost any gain or offset. For example, an input signal that varies from 2.00 to 2.15 volts or a signal that varies from 0 to 10 mV can easily be converted to 0-10V output for the A/D

The signal conditioner's outputs are on a high quality 16 pin gold I.C. socket that matches the one on the A/D's so a simple ribbon cable connects the two. The signal conditioner can be powered by your Apple or from an external supply

FEATURES

- 4" square for standard card cage and 4 mounting holes for standard mounting. The signal conditioner does not plug into the Apple, it can be located up to 1 mile away from the A/D
- 22 pin 156 spacing edge card input connector (extra connectors are easily available i.e. Radio Shack)
- Large bread board area
- Full detailed schematics included

PRICE \$79

I/O 32

- Provides 4, 8-8 bit programmable I/O Ports
- Any of the 4 ports can be programmed as an input or an output port
- All I/O lines are TTL (0-5 volt) compatible
- Your inputs can be anything from high speed logic to simple switches
- Programming is made very easy by powerful on-board firmware
- The I/O 32 is your best choice for any control application

The I/O manual includes many programs for inputs and outputs.

Some applications include:

Burglar alarm, door time sensing, use with relays to turn on lights, sound buzzers, start motors, control tape recorders and printers, use with digital joystick

PRICE \$89

Please see our other full page ad in this magazine for information on Applied Engineering's Intermaster I block I and other products for the Apple. Our boards are far superior to most of the consumer electronics made today. All IC's are in high quality sockets with no solder. Components used throughout P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products are compatible with Apple II and I/II.

Applied Engineering products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no-hesitate three year warranty.

Texas Residents Add 5% Sales Tax

Add \$10.00 if Outside U.S.A.

APPLIED ENGINEERING

P.O. Box 790
Carrollton, TX 75006

9 a.m. to 11 p.m. 7 days a week
Master card, Visa & C.O.D. Welcome
No extra charge for credit cards



SPECIAL !!! EXPANDED RAM/ROM BOARD: \$39.00

Similar to our \$30 RAM/ROM dev board described below. Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

IMPROVED !!! II IN A MAC (ver 2.0): \$75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

SCREEN.GEN: \$35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIC. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

MIDI-MAGIC for Apple //c: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple Imagewriter, Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

* **FONT LIBRARY DISKETTE #1: \$19.00** contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e : \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Grafrax-80, MX-100, MX-80/100 with Grafraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), Okidata 82A/83A with Okigraph & Okidata 92/93.

'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into \$Cn00-CnFF and \$C800-CFFF.

C-PRINT For The APPLE //c: \$69.00

Connect standard parallel printers to an Apple //c serial port. Separate P/S included. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange N.J 07052 (201) 325-1885



Display the 65802 Registers.....Bob Boughner

Shortly after I installed the 65802 chip in my Apple IIPlus, I inadvertently tried to execute a BRK instruction while in the 16-bit Native Mode. Needless to say, things did not work too well. After digging into the problem, I uncovered two areas where difficulties occur. The first is that the monitor ROMs implicitly assume 8-bit word lengths; going into them while in the 16-bit mode will definitely not work.

The second area, and even more serious, is that the BRK vector is located at a different place when you are in Native Mode. In Emulation Mode, or in a normal 6502 or 65C02, the BRK vector is combined with the IRQ vector and is at \$FFFE and \$FFFF. When you are in Native Mode in a 65802 or 65816, the BRK vector is separated from the IRQ vector. The Native Mode IRQ vector is at \$FFEE and \$FFEF, while the Native Mode BRK vector is at \$FFE6 and \$FFE7. A separate vector is required in Native Mode because there is no B-bit in the Native-mode status register.

After this experience, I decided to write a subroutine that would display all the registers, regardless of what mode I was in. The program which follows is the result of that effort as modified by Bob S-C. The program will only work in a 65802 or 65816; if your Apple has a 65C02 or 6502, you are out of luck because I used some instructions and addressing modes not in those processors. As you will notice, I decided not to bother with displaying the program bank and data bank registers, because in a 65802 these have no effect on anything. I also assume the program will be running in an Apple II machine below the IIgs level, so these registers will probably always be zero.

I also assumed in my code that the stack always stays in page one. This may not be a good assumption, but it simplifies a lot. Whenever a program switches to emulation mode, the high byte of the S-register gets changed to 01. Since I am using monitor subroutines, I have to use emulation mode. I further have assumed that the stack-relative addressing mode will work. This means that the stack cannot wrap around during the call and execution of DISPLAY.REGISTERS. If it does, the values printed out may not be correct.

I decided to split the display so that it will fit on a 40-column screen. The PC address and the A-, X-, and Y-registers display on the first line. The D- and S-registers display on the second line, followed by the nine status bits. I break out the status bits and display the bit name in normal video if the bit is 0. If a status bit is 1, its name will display in INVERSE on a //e or //c 80-column screen, and in FLASHING on a 40-column screen. By changing line 1650 to "AND #\$DF", and line 1990 to "FLAGS .AS -/eczidxmvmn/", you can make 0 status bits display in lower case and 1 status bits display in upper case (assuming your Apple has lower-case display).

Bob S-C made one change which saved a lot of code but will greatly offend some programmers. He added some self-modifying code! Line 1480 stores a value into the instruction in line

1490, building a "LDA ...,S" instruction with the proper offset to pick up a byte out of the stack.

Lines 1190-1220 are the standard way of entering a program which can be called from both Native or Emulation Mode. The first PHP saves the actual status register, and the second one saves the value that was in the E-bit. The CLC-XCE gets us into Native Mode, so that we can save the registers. Since we might have been in full 16-bit native mode, we have to be in that mode when we save the A-, X-, and Y-registers.

Lines 1230-1270 save the registers by pushing them on the stack. Lines 1290 set the D-register to 0000, so that we can call monitor subroutines later. Lines 1320-1350 form a modified return address, so that we can print out the actual address of the JSR DISPLAY.REGISTERS instruction. This really is not that necessary, but I liked it better this way. Lines 1370-1400 push the address on the stack of the stack pointer before the JSR DISPLAY.REGISTERS was executed. Now there are quite a few bytes on the stack, so I will draw a map. The numbers indicate the value that can be used in a "offset,S" addressing mode to access the bytes.

```
$10,S  return-hi
$0F,S   "    -lo
$0E,S   P-register (status)
$0D,S   "    (E-bit)
$0C,S   D-register-hi
$0B,S   "    -lo
$0A,S   Y-register-hi
$09,S   "    -lo
$08,S   X-register-hi
$07,S   "    -lo
$06,S   A-register-hi
$05,S   "    -lo
$04,S   call addr-hi
$03,S   "    -lo
$02,S   stack addr-hi
$01,S   "    -lo
```

Lines 1420-1550 display the register contents of the 16-bit registers. A format string controls the loop. The bytes in the string which are ASCII characters with the high bit set are printed just as they are. Bytes in the format string which are hex values below \$20 are offsets into the stack. These are used to pick up bytes out of the stack for printing in hexadecimal.

Lines 1560-1690 display the nine status bits. I first shove the E-bit value into CARRY, then pick up the rest of the bits in the A-register. The loop then shifts A and CARRY, so that I can effectively squeeze nine bits into the eight-bit A-register.

Lines 1710-1830 restore all the registers and return to the caller in the same mode and with the same status as at the time of call.

I added a little test routine to the end, lines 2010-2100.
This loads a few test values into registers in native mode and displays them. The result looked like this:

```
0898-  A=1234  X=5678  Y=ABCD
        D=0000  S=01E2  NVMXDIZCE
```

with N and C in inverse or flashing mode.

When I called DISPLAY.REGISTERS directly by typing \$800G from within the ProDOS version of the S-C Macro Assembler, the display looked like this:

```
8B52-  A=00DC  X=0002  Y=0000
        D=0000  S=01E2  NVMXDIZCE
```

with M, X, and E in inverse or flashing mode.

I use my DISPLAY.REGISTERS subroutine when I am debugging a new program, by assembling it in and calling it at strategic points. I am working on a more complete DEBUG program which will allow STEP and TRACE for all of the 65802 opcodes. I am using Bob S-C's DISASM.816 (published in AAL in March 1985 and March 1986), and Steve Wozniak's 6502 TRACE from the old monitor ROM.

```

1000  *SAVE S.REG.DISP
1010  *-----
1020  .OP 65802
1030  *-----
1040  MON.PREYTE .EQ $FDDA
1050  MON.COUT  .EQ $FDED
1060  *-----
1070  * SUBROUTINE TO DISPLAY 65802 REGISTERS
1080  * By Bob Boughner and Bob S-C
1090  *
1100  * I assume this routine will be used in
1110  * an Apple with only 64K address space,
1120  * so PBR and DBR are always 00.
1130  *
1140  * I also assume that the S-register is
1150  * always in page 1, because it must be
1160  * there for monitor subroutines to work.
1170  *-----
1180  DISPLAY.REGISTERS
1190  PHP          SAVE CALLER'S STATUS
1200  CLC          ENTER NATIVE MODE
1210  XCE
1220  PHP          SAVE CALLER'S E-BIT
1230  REP #$38    CLR M,X,D (16-BIT, NOT DECIMAL MODE)
1240  PHD          SAVE D-REGISTER
1250  PHY          Y-REGISTER
1260  PHX          X-REGISTER
1270  PHA          A-REGISTER
1280  *-----
00080A- A9 00 00 1290  LDA ##0000  D=0000, so no problems with use of monitor
00080D- 5B          1300  TCD          subroutines.
1310  *-----
00080E- A3 0B 1320  LDA 11,S    Get and adjust Return Address to point to
000810- 3A          1330  DEC          the JSR that got us in here.
000811- 3A          1340  DEC
000812- 48          1350  PHA
1360  *-----
000813- 3B 1370  TSC          Get and adjust Stack Pointer
000814- 38 1380  SEC
000815- B9 0D 00 1390  SBC ##0013
000818- 48 1400  PHA
1410  *-----
000819- 38 1420  SEC          Get into emulation mode for display
00081A- FB 1430  XCE
1440  *---Display 16-bit registers---
1450  LDY #0
00081D- B9 59 08 1460  .1 LDA FORMAT,Y
000820- 30 0A 1470  BMI .3
000822- 8D 26 08 1480  STA ,2+1 *** SELF-MODIFYING CODE ***
000825- A3 00 1490  .2 LDA =#,S

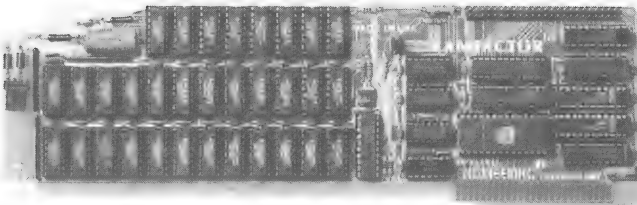
```

RamFactor™

The Highest Performance, Speed and Software Compatibility of Any Slot 1-7 Memory Card for the Apple II+, IIe or IIGs

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIGs preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and up to 16 MEG with additional piggyback card. And since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option you can switch on your Apple and any of your favorite software will be up and running in less than 1 second!



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PPS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDOS, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDOS clock. RamFactor will automatically segment large files so they can be saved on 5¼", 3½", and hard disks. All this performance is available to anyone with an

Apple IIe, IIGs or II+ with an 80 column card. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

True 65C816 16 Bit Power

RamFactor has a built-in 65C816 CPU port for direct connection to our IIe 65C816 card for linearly addressing up to 16 MEG for the most powerful 16 bit applications (II+ 65C816 card under development)

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDOS in under a second. And with our Battery back-up option, you can have permanent storage for up to 20 years.

Quality and Support of the Industry Leader

RamFactor is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

Features:

- Up to 16 MEG total memory, 256K to 1 MEG on main board. Up to 16 MEG with additional memory on piggyback card
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIGs, Franklin and Laser 128
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDOS, DOS 3.3, Pascal and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in linear addressing 16 bit co-processor port
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or later
- Allows Apple II+ and IIe to run your AppleWorks without buying additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDOS clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$289
RamFactor with 1 MEG	\$389
RamFactor with 2-16 MEG	CALL
Battery Back-up Option	\$179
65C816 16 Bit Card	\$159

Order RamFactor today... with 15 day money back guarantee and our "no hassle" five year warranty. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 5 1/4% sales tax. Add \$10.00 if outside U.S.A.

AE Applied Engineering™
The Apple enhancement experts
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

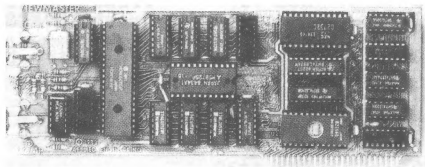
```

000827- 20 DA FD      1500      JSR MON.PRBYTE
00082A- 80 03      1510      BRA .4
00082C- 20 ED FD      1520      JSR MON.COUT
00082F- C8      1530      INY
000830- C0 29      1540      CPY #FORMAT.LEN
000832- 90 E9      1550      BCC .1
000834- A0 08      1560      *---Display status bits-----
000836- A3 0D      1570      LDY #8      9 bits altogether
000838- 4A      1580      LDA 13,S      Get E-bit into CARRY
000839- A3 0E      1590      LSR
00083B- 2A      1600      LDA 14,S      Get the rest of the bits
00083C- 48      1610      ROL      First time this puts E into bit0
00083D- B9 82 08      1620      PHA      Save status byte
000840- 90 02      1630      LDA FLAGS,Y      Get bit name
000842- 29 7F      1640      BCC .6      Print NORMAL if bit=0
000844- 20 ED FD      1650      AND #$7F      Print INVERSE or FLASH if bit=1
000847- 68      1660      JSR MON.COUT
000848- 88      1670      PLA      Get status byte again
000849- 10 F0      1680      DEY      Next bit
00084B- 18      1690      BPL .5
00084C- FB      1700      *---Restore everything & return--
00084D- C2 30      1710      CLC      NATIVE MODE
00084F- 68      1720      XCE
000850- 68      1730      REP #$30
000851- 68      1740      PLA      POP OFF S-REG VALUE
000852- FA      1750      PLA      POP OFF PC VALUE
000853- 7A      1760      PLA      RESTORE A-REG
000854- 2B      1770      PLX      X-REG
000855- 28      1780      PLY      Y-REG
000856- FB      1790      PLD      D-REG
000857- 28      1800      PLP      E-BIT
000858- 60      1810      XCE
000859- 04 03      1820      PLP      STATUS
00085B- AD A0 A0 C1      1830      RTS
00085F- BD      1840
000860- 06 05      1850      *-----
000862- A0 A0 D8 BD      1860      FORMAT .HS 04.03      PC of JSR DISPLAY.REGISTERS
000866- 08 07      1870      .AS -/- A=/
000868- A0 A0 D9 BD      1880      .HS 06.05      A-register
00086C- 0A 09 8D      1890      .AS -/ X=/
00086F- A0 A0 A0 A0      1900      .HS 08.07      X-register
000873- A0 A0 A0 C4      1910      .AS -/ Y=/
000877- BD      1920      .HS 0A.09.8D      Y-register, RETURN
000878- 0C 0B      1930      .AS -/ D=/
00087A- A0 A0 D3 BD      1940      .HS 0C.0B      D-register
00087E- 02 01      1950      .AS -/ S=/
000880- A0 A0      1960      .HS 02.01      S-register
000882- C5 C3 DA C9      1970      .AS -/ /
000886- C4 D8 CD D6      1980      *-----
00088A- CE      1990      FORMAT.LEN .EQ *-FORMAT
00088B- 18      2000      *-----
00088C- FB      2010      T
00088D- C2 30      2020      CLC
00088F- A9 34 12      2030      XCE
000892- A2 78 56      2040      REP #$30
000895- A0 CD AB      2050      LDA ##$1234
000898- 20 00 08      2060      LDX ##$5678
00089B- 38      2070      LDY ##$ABCD
00089C- FB      2080      JSR DISPLAY.REGISTERS
00089D- 60      2090      SEC
00089E- 60      2100      XCE
00089F- 60      2110      RTS
0008A0- 60      2120      *-----

```

Viewmaster 80

Now run AppleWorks™ on your II+ *and* keep full 80 column compatibility!



	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE	800 IN SOFTWARE
VIEWMASTER 80	x	x	x	x	x	x	x	x	x
SUPRTERM									
WIZARD 80									
VISION 80	x	x							
OMNIVISION									
VIEWMAX 80	x	x							
SMARTER									
VIDEOTERM									

One look at the chart will give you some of the reasons there's only one smart choice in 80 column cards for your Apple. But the real secret to Viewmaster 80's success is something even better: Total compatibility.

Each Viewmaster 80 includes our powerful AppleWorks expand software, allowing AppleWorks to run on the II+ with only 64K (or more) memory. (We recommend the RamFactor memory card, but any compatible Apple memory card will work.) The software provides our full range of AppleWorks enhancements, including expanded records, word processor, multi-disk saving, time and date display on screen with any PRO-DOS clock, and more!

The Viewmaster 80 works with all 80 column applications, including DOS 3.3, PRO-DOS, CP/M, Pascal, WordStar, Format II, Applewriter II, dBase II, Visicalc, Multiplan, and *hundreds* of others—including AppleWorks.

Here are just a few of the powerful features the Viewmaster 80 delivers for a great price (\$149):

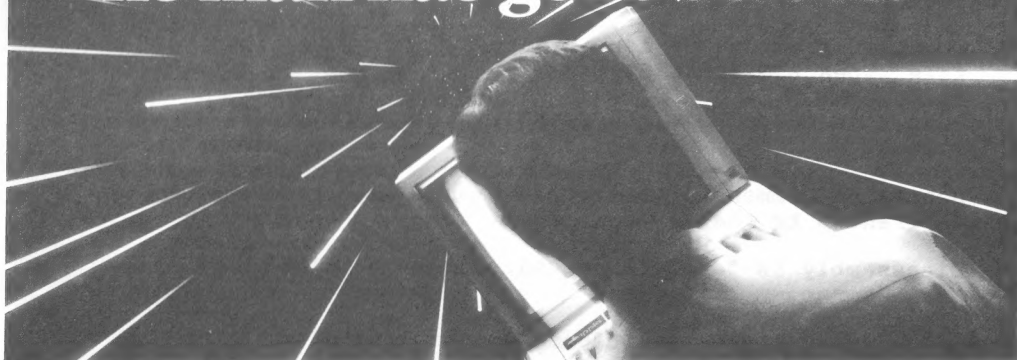
- 80 Characters by 24 lines • Fully compatible with all Apple languages and software • Highest compatibility with existing 80 column software • Very low power consumption • High speed (18 MHZ) scroll rate • Upper and lower case characters and true descenders, both inverse and normal; all on-screen editing functions are supported • User-definable cursor shape • Compatible with Apple II, II+ and IIe • Five-year warranty

Call today to order or for more information, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

CP/M is a registered trademark of Digital Research, Inc.

AE Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

To boldly go at speeds no man has gone before.



*With the TransWarp™ accelerator, your IIe™ or II+™
is 40% faster than the IIGs!*

Computing at warp speed!

It's an experience you shouldn't miss. And with TransWarp, you won't have to. Because TransWarp will run your IIe or II+ software *3.6 times faster*—that's 40% faster than Apple's IIe's!

No more yawning while your Apple™ slowly rearranges text or calculates spreadsheets. With 256K of ultra-fast RAM, TransWarp speeds up *all* Apple software—including AppleWorks, and all educational software, graphics and games. And it's compatible with all standard peripheral cards (such as RamWorks and Apple memory cards), hard disks, 3 1/2" UniDisks, 80-column cards, modems, clock cards, mouses and more! You name it, TransWarp accelerates it. There's even a 16 bit upgrade chip available.



*"TransWarp is great! I have
replaced all my other
accelerators with it!"*

*Steve Wozniak, the creator
of Apple Computer*

An important difference.

With TransWarp, software runs up to three times faster than with other cards, since the others can't accelerate programs in auxiliary memory. That's why TransWarp is so much faster than the rest. Nearly all of today's more powerful programs run partially or completely in auxiliary memory: programs like AppleWorks, Pinpoint, Managing Your Money, SuperCalc 3a, BPI and Pascal, to name a few. Why settle for a card that accelerates only part of the memory?

There's one more important difference. Since TransWarp doesn't use memory caching, you get *consistent* high speed performance.

A cinch to use.

Simply plug TransWarp into any slot in your Apple II, II+ or IIe—including slot 3 in the IIe. Instantly you'll be computing

at speeds you only dreamed about before. And should you ever wish to run at normal speed, simply press the ESC key while turning your Apple on.

Since TransWarp is completely transparent, you won't need preboot disks or special software. It's ready to go right out of the package!

Speed = Productivity

Imagine the productivity gains you'll achieve when your programs are running over three times faster. TransWarp is so powerful, your Apple will make IBM PCs™ and even ATs™ look like slowpokes.

- 3.6 MHz 65C02
- 256K of ultra-fast on-board RAM
- Accelerates main *and* auxiliary memory
- Low power consumption for cool operation
- Totally transparent operation with all software
- Plugs into any slot, including slot 3 on the Apple IIe
- Accelerated 16 bit option available
- 5-year warranty

Satisfaction guaranteed!

Give your Apple the TransWarp advantage. With our risk-free 15 day money-back guarantee, you have nothing to lose but wasted time. Call today!

TransWarp accelerator
16 bit upgrade (may add later)

\$279
\$89

For fast response:

Call Applied Engineering, 9 a.m. to 11 p.m., 7 days at (214) 241-6060. MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 if outside U.S.A. Or mail check or money order to Applied Engineering, P.O. Box 798, Carrollton, TX 75006.

AE Applied Engineering
The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006 (214) 241-6060

In response to the requests of many of you, I have at long-last developed a disassembler which runs under ProDOS. (This new product is distinctly different from the Rak-Ware DISASM, which runs under DOS 3.3. The Rak-Ware product is still the best one to use if you are using DOS.) Here are some of the features of the new S-C ProDOS DISASM:

- * Input is from one or more binary object files, including file types BIN and SYS.
- * Output is to one or more "S-C" type (compressed source code) files.
- * Generates comment lines before each label listing all references to that label.
- * Disassembly is "script" driven, allowing incremental enhancement.
- * Input files may be positioned to specific starting addresses.
- * Decodes ProDOS "MLI" calls as such.
- * Allows pre-named symbols up to 32-characters long.

Of all the features, the most important may be the "script". This is essentially a "program", written in "disassembly language". The script allows you to define which input files to include and which output files to generate, to name symbols such as monitor entry points and major subroutines in your program being disassembled, to define table areas, and even to insert comments.

The script itself is written using the standard S-C Macro Assembler, and may be saved on a source file just as an assembly-language program would. As you gain knowledge about the program you are disassembling, you can add lines to the script.

Version 1.0 handles all of the 65C02 instruction set. Future enhancements which are definitely planned include expanding to include the entire 65816 instruction set. Version 1.0 is for sale now for \$50 including the commented source code.

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are \$1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)